PCT/IB05/50399

REC'D 03 MAR 2005
WIPO PCT

PA 1278465

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

January 31, 2005

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: *60/541,206*
FILING DATE: *February 02, 2004*

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

P. SWAIN
Certifying Officer

14230 U.S. PTO

322782 U.S. PTO
60/541206
020204

Please type a plus sign (+) inside this box → +

PTO/SB/16 (02-01)
Approved for use through 10/31/2002. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request f r filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

Express Mail Label No. EV 312 068 343 US **Date f D p sit: 02 February, 2004**

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Nevenka	Dimitrova	Yorktown Heights, NY
Jun	Fan	Shenzen, Guangdong Prov, CN

☐ *Additional inventors are being named on the _____ separately numbered sheets attached hereto*

TITLE OF THE INVENTION (280 characters max)

CONTINUOUS FACE RECOGNITION WITH ONLINE LEARNING

CORRESPONDENCE ADDRESS

Direct all correspondence to:

☒ Customer Number 24737 → *24737*

OR *Type Customer Number here*

☒ Firm *or* Individual Name	PHILIPS INTELLECTUAL PROPERTY & STANDARDS
Address	345 SCARBOROUGH ROAD
Address	P. O. Box 3001
City	BRIARCLIFF MANOR, State NY, ZIP 10510
Country	USA, Telephone (914) 333-9611, Fax (914) 332-0615

ENCLOSED APPLICATION PARTS *(check all that apply)*

Specification *Number of Pages* ☐ CD(s), Number

☐ Drawing(s) *Number of Sheets* ☒ Other (specify) ID# 779725

☐ Application Data Sheet. See 37 CFR 1.76

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check or money order is enclosed to cover the filing fees

FILING FEE AMOUNT ($)

☒ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 14-1270 — 160.00

☐ Payment by credit card. Form PTO-2038 is attached.

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____.

Respectfully submitted,
SIGNATURE

Date 2/2/2004

REGISTRATION NO.: 28,613
(if appropriate)

TYPED or PRINTED NAME EDWARD W. GOODMAN

Docket Number: US040102

TELEPHONE (914) 333-9611

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C., 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Philips Intellectual Property & Standards

Invention Disclosur
Version 1.4.0 14-11-2002

This template is meant for making a detailed description of your invention off-line. The description can then later be attached in electronic form to your ID submission. You must also use this off-line description to obtain export clearance from your local export-control officer.

Inventors name(s) + e-mail address(es):

Jun Fan Jun.Fan@philips.com
Nevenka Dimitrova, Nevenka.Dimitrova@philips.com

Title: Continuous face recognition with online learning

WARNING: Applicable to United States of America, China and United Kingdom. In these countries export control is applicable. Please use this ID template and get clearance from your local export control officer before sending your idea via Inventions On Line to IP&S.

For Export Control Officer (ECO):
Before this ID can be submitted to the automated Invention On Line System of Philips Intellectual Property & Standards it is sent to you in your capacity of Export Control Officer of the entity (Research/Business Unit) the inventor(s) belong(s) to in order to obtain your legally required export clearance. Please fill out the following fields and return the completed form to the inventor/sender.

Name ECO:Betsy McIlvaine

Research/ Business Unit:Philips Research USA

Export clearance yes/no:yes

Export control clearance number:NLR EAR99

Letter of assurance required? (US):no

Any other comments:

Date of export control:28 Sept 03

In case of any questions ask *IP&S Helpdesk ICT*

Detailed description of your invention:
Please describe your invention in terms of what is already known, the measures/features of your invention and the resulting advantages. The following scheme may be useful.

a. Give some background information in the field of your invention, i.e. state of the art, such as commonly known facts, devices etc. If possible, state references to public documents, such as articles in technical journals, proceedings of conferences, brochures, or patent documents.

Digital image and digital video are becoming more and more popular currently for both home users and business users, and Facial analysis has been developed very fast recently. Face recognition for faces that are known in advance exists.

b. Indicate problems/disadvantages with the state of the art that your invention will overcome. Are the disadvantages/problems new or were they already known?

The problem with face recognition systems today is that they are not able to recognize and learn new faces automatically while in operation.

c. The measures/device features that are proposed to solve the problem, and the resulting advantages. If the invention is based on a new understanding (insight), please indicate this.

The proposed system in this invention automatically adds new faces to an existing database and keeps learning new faces. So, in that sense, in contrast to existing systems our online learning system can learn features of new faces and store corresponding models for new faces. This is especially important in the area of new digital cameras and PDAs and portable storage containers with imaging capability.

d. Provide at least one embodiment of the invention, where you explain the best way of carrying out the invention. Please add drawings, graphs, test data etc. where appropriate.

Summary
The main idea is to recognize known faces, detect unknown faces and apply automatic online learning for unknown faces in videos. After the online learning, our Classifier could recognize the new (unknown) faces presented before. After the recognition, the Classifier will assign recognized face IDs to the faces.

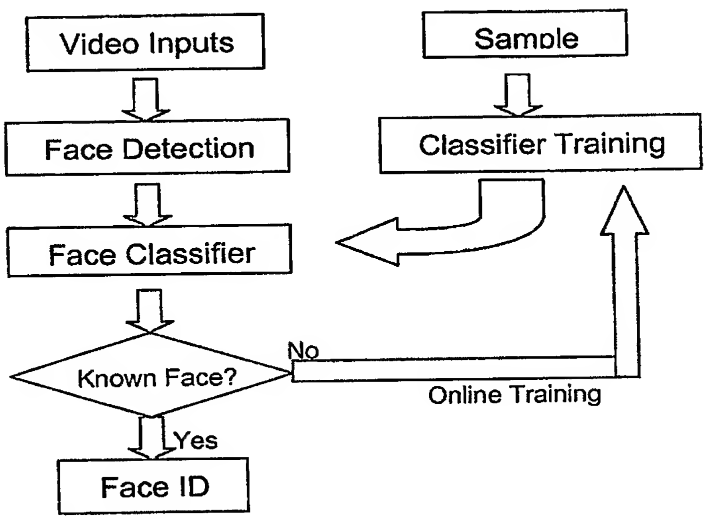
Acquisition
The faces using for classifier training and online learning could be found on Internet sources or screenshot from the video. After our classifier detect a new face, the database will be updated with the new faces.

What's new?
Most face recognition system will only recognize a fixed number of faces in the database and the face database cannot be updated during the classifying procedure. Our approach could automatically detect the new faces and extend the database based on the new faces. Also, our approach could generate a confidence measurement for each recognized face in the database and sort the candidate by the confidence measurement, which make post-processing easier.

Modified Probabilistic Neural Networks
Probabilistic Neural Networks are used in the standard machine learning literature. The purpose of the modification on Probabilistic Neural Networks is to detect an unknown pattern and to do online learning for the unknown pattern so that the unknown pattern could be recognized next time by our networks. In order to detect the unknown pattern, a threshold is set on the category layer of Probabilistic Neural Networks. The threshold could be

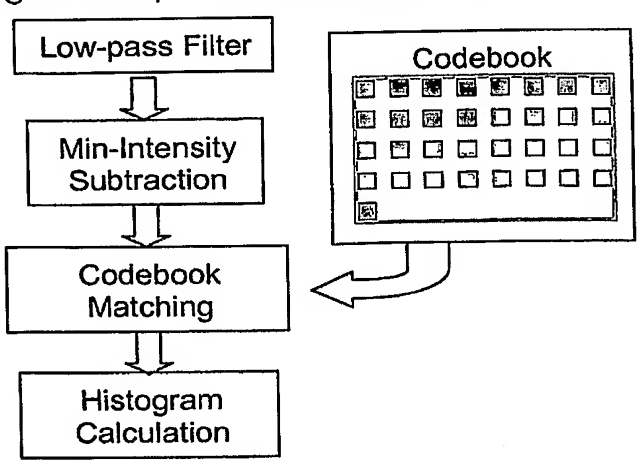
comparative or absolute value depending on how the system is designed. If the output value from the hidden layer is lower than the threshold, the Classifier will recognize the input pattern as unknown pattern. After it detects an unknown pattern, the Classifier will store the unknown pattern information in hidden layer. Therefore, when the unknown pattern appears next time, the Classifier could recognize this unknown pattern.

System Diagram



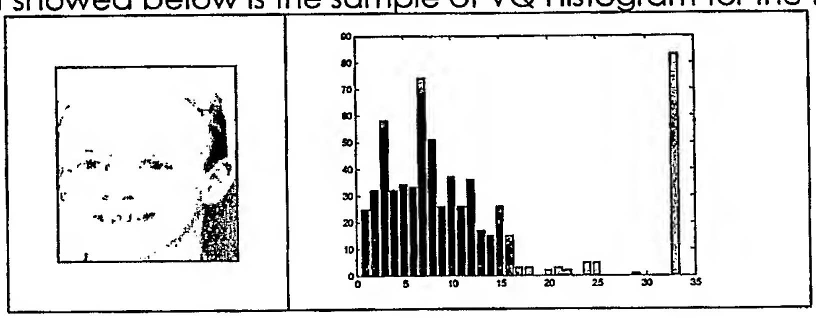
Face Features

In our implementation, we choose Vector Quantization (VQ) Histogram feature for the Classifier.However, one can use another feature space that is used for face recognition in the literature (e.g. EigenFaces). The VQ Histogram calculation procedure is showed below:



We forward the face image into a low-pass filter first. The low-pass filter is used for reducing the high-frequency noise and extracting the most effective low frequency component for recognition. We then divide the image into 4-by-4 block. Next, calculate the minimum intensity in each 4-by-4 pixel block, and subtract the minimum intensity for each block. Therefore, we could get an intensity variation in each block. Then, for each block division from the face image, we match the block with all the codes in the codebook, and the most similar-matched codevector is selected. Euclidean distance is used for the distance matching. After performing VQ for all the blocks divided from a facial image, matched frequencies for each codevector are counted and a histogram is generated.

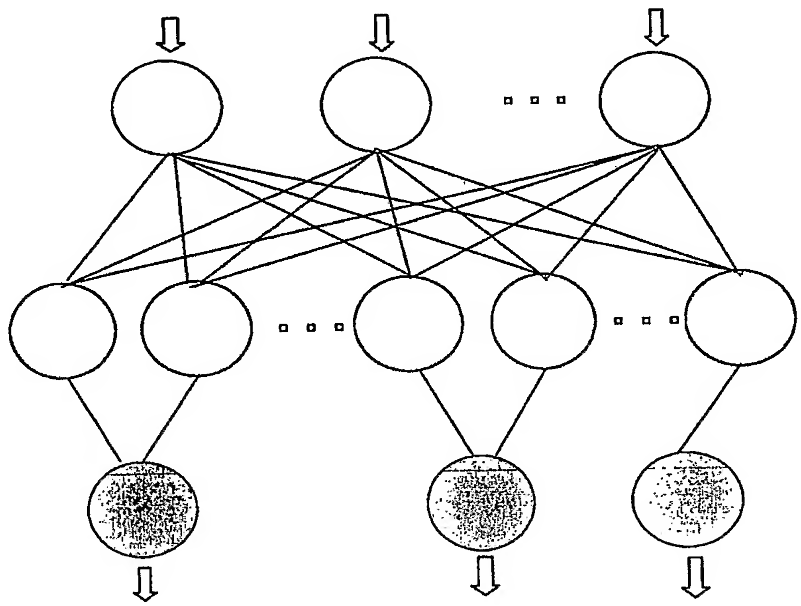
The figure on the right showed below is the sample of VQ Histogram for the face on the left.



The VQ Histogram is insensitive to geometry information of the face, robust to lighting, posing and expression and also independent of face position if the background is uniform.

Probabilistic Neural Networks Structure

The standard Probabilistic Neural Networks contains three layers: Input layer, pattern layer and category layer. The Input Layer will normalize the input vectors and forward the normalized data to Pattern Layer. Pattern layer will then add one node in pattern layer for each training and save the weights as normalized input vectors. The category layer will choose the maximum value calculated from Pattern Layer and calculate the confidence measurement based on the values from pattern layer. The Probabilistic Neural Networks diagram is showed below:



Probabilistic Neural Network Training
The probabilistic Neural Networks will normalize all the data from input vector. The normalization calculation is shown below:

$$x'_n = \left(\frac{1}{\sum x_n}\right) \circ x_n$$

X_n is the input vector and $X`_n$ is the normalized input vector.

For each training data, probabilistic Neural Networks add one node in pattern layer and link the node to all the nodes in input layer. It will assign the normalized input vector as weights between the link from input layer and pattern layer. Then, link the new node in hidden layer to the supervised category.

Classifying Process using Probabilistic Neural Networks
During the Classifying process, the probabilistic Neural Networks will also do the normalization as showed on probabilistic Neural Networks Training. Then in the pattern layer, the normalized input data will perform a dot product to the saved weights as showed below:

$$Z_i = X \cdot W_i$$

The X is the normalized input vector and W_i is the weight. Z_i is the dot product value for each node in pattern layer.

Then do the following calculation:

$$Z'_i = \sum_i^{i \in category_k} \exp[(Z_i - 1)/\sigma^2]$$

The σ is the smooth factor and Z'_i is the output value for each node in pattern layer.

In category layer, The Probabilistic Neural Networks will compare the output value from pattern layer and calculate the confidence measurement based on the value from pattern layer. The nodes in category layer will do the following calculation:

$$C_i = \frac{(Z'_i / n_i)}{\sum Z'}$$

The C_i is the confidence measurement for each category, n_i is number of pattern nodes for i category.

Online Training & Detecting new faces

We use a fuzzy determination to detect the new coming faces. The fuzzy determination is performed using the following rules:

1. Output value from pattern layer is below threshold
2. Mean of output value from pattern layer during a time series is low
3. Distance to other clusters is stable
4. The face lasted for a particular time slice

After we detect a new coming face, we will do the online training in order to store the information of the new face. The online training will do:

1. Add new nodes in hidden layer for each new face
2. Do the normalization for the new face's input vector and save them as weights
3. Link the new nodes to a new category in category layer

The information of new face will be stored in hidden layer and category layer of Probabilistic Neural Networks. Therefore, when the "new face" appear again, the probabilistic Neural Networks will recognize this face as known face.

Implementation

There are several possible implementations as described below

∘Actor/Actress recognition

Our face recognition system could recognize the faces in the movies, home videos and business videos with a pre-trained Probabilistic Neural Networks. For the face recognition in the movies, we could find all of the player names from the screenplay or other sources (e.g. IMDB.com). Search all of the players' faces on Internet, and train the Probabilistic Neural Networks based on the images found on the Internet. After the training, the Probabilistic Neural Networks is ready for classifying then.

∘Automatic editor for home movies

In this scenario

Our user wants to see the movie clip only when some specific persons appear in the video. Applying our face recognition system could solve this problem. The system will recognize faces in the movie and extract the clip based on the recognized faces. If the recognized faces are the faces that the user wants to see, then we will show user this clip.

∘Role Analysis

Apply face recognition system for the whole video and count the appearance time for each recognized face. The face got longest appearance time is the main actor/actress of the video.

∘Automatic Photo Distribution: Automatic picture e-mailing system based on face recognition.

∘Search Photo based on the wanted people

The face recognition system could also be applied in images. For example, our user have a huge digital image library, and he/she want to look at his/her grandma's pictures. Our face recognition system could help user to do the time consuming search. After we trained the Probabilistic Neural Networks based on the persons we want to recognize, we apply the Probabilistic Neural Networks to all the images in the library. Our system will automatically recognize his/her grandma and show the images to user.

∘Meeting Summarization

Combining with information from other sources (e.g. Audio, Meeting notes), we could generate a meeting summarization associated with the face ID, meeting notes and time stamp.

e. Indicate in which fields (technical, commercial) the invention can be applied.

Digital cameras, PDAs mobile phones with cameras, home media server, DVD+RW combi product.

Online Face Recognition System based on Modified Probabilistic Neural Networks with Adaptive Threshold for Videos

Jun Fan, Nevenka Dimitrova and Vasanth Philomin
Philips Research USA
345 Scarborough Rd.
Briarcliff Manor, NY 10510

Abstract

We developed an online-learning face recognition system based on the Modified Probabilistic Neural Networks (MPNN) for videos. This face recognition system can detect and recognize face, as well as automatically detect unknown faces and train the unknown faces online into face Classifier so that this "unknown face" can be recognized if it appears again. The MPNN is implemented by setting threshold on the category layer of Probabilistic Neural Networks (PNN) in order to detect unknown category of input data. Following some fuzzy rules based on the detected unknown category from MPNN, the system could then detect the unknown faces in videos. The PNN feed-forward training makes the online training very fast without changing the weights between the trained faces.

1. Introduction

Digital images and digital video are becoming more and more ubiquitous currently for both home users and business users. Most of these still and moving images contain people and their activities. As the number of these images explodes, it becomes a survival task to access these images based on who is present. Facial analysis has been a very active research area recently. Face recognition for faces that are known in advance exists. The biggest weakness of face recognition systems today is that they are not able to recognize and learn new faces automatically while in operation.

Most face recognition systems are trained on a fixed number of faces which are known in advance. These systems will only recognize the faces with known models and the face database cannot be updated during the classification procedure. In this respect these systems are very limited once they are placed in operation. They will work for surveillance systems, which have to recognize all employees of a company and alert to any intruders. However, in the area of home video, TV broadcast video, wearable video, there are new people appearing as the story unfolds. If a system is trained to recognize only family members then a visitor is labeled as "other" or "unknown". Of course there are travel videos with many new faces that are transient. A system that categorizes images and videos based on people presence has to distinguish all these categories of important and unimportant faces. Moreover, the system has to be flexible enough to incorporate and retain important faces.

Our approach can automatically detect the new faces and extend the database based on the new faces. Our online learning system can learn features of new faces and store corresponding models of new faces for future use. Also, our approach can generate a confidence measurement for each recognized face in the database and sort the candidate by the confidence measurement, which make post-processing easier.

2. System Architecture

Figure 1~~Figure 1~~ shows our face recognition system architecture. There are two approaches to bootstrap the system: 1) Initial database has a limited number of faces, and 2) Initial database is empty. If the system is first trained on the initial database, we gain high recognition accuracy. This method is similar to our human perception of known faces and incorporation of new faces. We assume that we know some faces of the video clip before we perform the online learning for the whole video clip. The system has a training phase and classification phase just like any other face

recognition system. However, the important aspect here is that there is a feedback arrow to the training face for unknown faces.

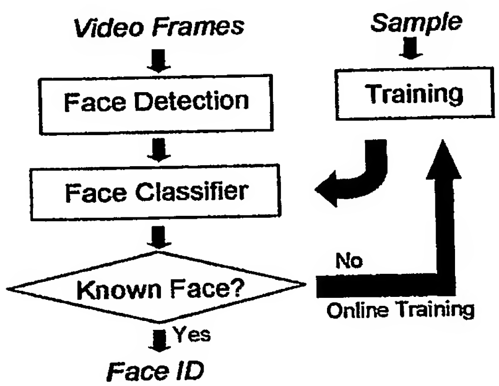


Figure 1. Face Recognition System Architecture

During the training phase, the system will read face examples for each face (actor/character) and train the Probabilistic Neural Networks (PNN) [4][2] based on the features of these faces. The size of PNN will increase during the training and it is decided by the following equation:

$$size = \sum_{i=1}^{m} n_i \qquad (1)$$

where n_i is the number of training faces for the person i and the m is the number of persons in the initial database to be trained by PNN.

During classification phase, the system will decode the MPEG video file into video frames first. For each frame, we use a variant of the face detector described in [8]. If there is a face found by the face detector, the face segment will be forwarded to the PNN based Face Classifier. If it is a known face, the confidence measurement for each face ID will be generated by PNN; otherwise, the unknown face will be evaluated and forward to online learning phase if necessary. After we have the confidence measurement for each Face ID, we can easily choose the Face ID with the maximum confidence measurement as the output from Face Classifier.

3. Face Detection

This section briefly describes the face detection algorithm used in our framework. In [8], Viola and Jones applied the popular AdaBoost [12] learning technique to the problem of rapid object detection. They used an attentional cascade of strong classifiers that consisted of a set of computationally efficient binary features (also called weak classifiers). Each round t of boosting added a single feature h_t to the current set of features by minimizing:

$$Z_t = \sum_i D_t(i) \exp(-\alpha_t y_i h_t(x_i)),$$

where $D_t(i)$ is the weight on example x_i at round t, $y_i \in [-1, 1]$ is the target label of the example, α_t is the influence of this weak hypothesis on the strong classifier and $h_t()$ is the weak binary hypothesis restricted to [-1, 1]. In our variant, we use boosting stumps (decision trees that partition the domain into two pieces and yield a prediction for each partition) as the weak classifiers and our goal is to now minimize:

$$Z_t = \sum_i D_t(i) \exp(-y_i h_t(x_i)),$$

where α_t has been folded into h_t, thereby allowing the weak hypotheses to have a range over all $\Re$ rather than the restricted range [-1, +1]. The prediction values for the left and right partitions that minimize Z_t above are:

$$c_{left} = \frac{1}{2}\ln\left(\frac{W_+^{left} + \varepsilon}{W_-^{left} + \varepsilon}\right); c_{right} = \frac{1}{2}\ln\left(\frac{W_+^{right} + \varepsilon}{W_-^{right} + \varepsilon}\right),$$

where the W's denote the weight of the examples that are assigned to the left or right partition with true labels "positive" or "negative". The predictions are also smoothed with the term ε to avoid numerical problems caused by large predictions. Typically, ε is chosen on the order of the reciprocal of the number of training samples in our system. From these prediction values, we can greedily choose the splitting criterion for the decision tree (dropping the subscript t)

$$\text{as } Z = 2\left(\sqrt{W_+^{left} W_-^{left}} + \sqrt{W_+^{right} W_-^{right}}\right)$$

rather than the Gini index or an entropic function [12].

A few variants [9][11] of the learning algorithm described in [8] have been proposed recently. These algorithms reduce the training error (i.e. error in the training set) during training and count on the generalization performance of AdaBoost that is rigorously proved in [12]. It is our experience that using a validation set during training as in [8][10] yields the most effective cascades. In addition, we just scan the validation set once (rather than several times as in [10]) for each weak classifier that is added to the current cascade in order to adjust the strong classifier threshold. We do this by keeping track of the rectangles and their corresponding last stage sums that pass through all but the penultimate stage of the current cascade (for the first stage, this amounts to

keeping track of all rectangles scanned and their corresponding sums). Our final trained cascades typically have around 30 stages and the entire training process takes slightly less than a week to complete on a dual 2.8Ghz Xeon processor with 2GB of memory. We use around 4000 positive samples and 5000 negative samples for training each stage of the cascade where the negative samples for each stage are the false positives obtained by scanning the current cascade on an image set with no faces. Our validation set consists of around 200 faces. The face detector runs comfortably in real-time on a 750Mhz P-3 laptop and can detect faces at 10 different scales.

4. Face Features

This section introduces a Vector Quantization (VQ) Histogram based face feature [1], which we chose for our Face Recognition system. However, one can use another feature space that is used for face recognition in the literature (e.g. EigenFaces [7]). The VQ Histogram calculation procedure is showed in Figure 2~~Figure 2~~.

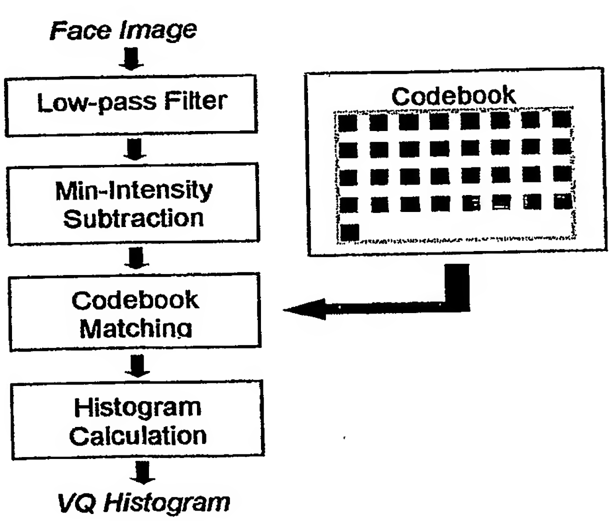


Figure 2. Face Feature Generation Procedure

In order to generate the VQ Histogram, we forward the face image into a low-pass filter first. The low-pass filter is used for reducing the high-frequency noise and extracting the most effective low frequency component for recognition. We then divide the image into 4-by-4 block. Next, we calculate the minimum intensity in each 4-by-4 pixel block, and subtract the minimum intensity from each pixel in the 4-by-4 block. Therefore, we can obtain an intensity variation in each block. Then, for each block division from the face image, we match the block with all the codes in the codebook, and the most similar-matched codevector is selected. Euclidean distance is used for the distance matching. After performing VQ for all the blocks divided from a facial image, matched frequencies for each codevector are counted and a histogram is generated.

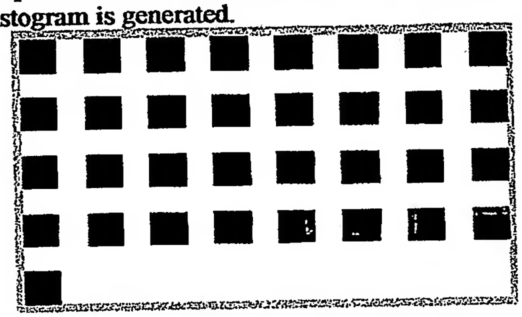


Figure 3. Organization of Codebook

The Organization of the codebook we used in our implementation is systematically organized with 33 codevectors having monotonic intensity variation. The first thirty-two vectors are generated by changing directions and range of intensity variation as show in Figure 3~~Figure 3~~. The last vector contains no directions and variation.

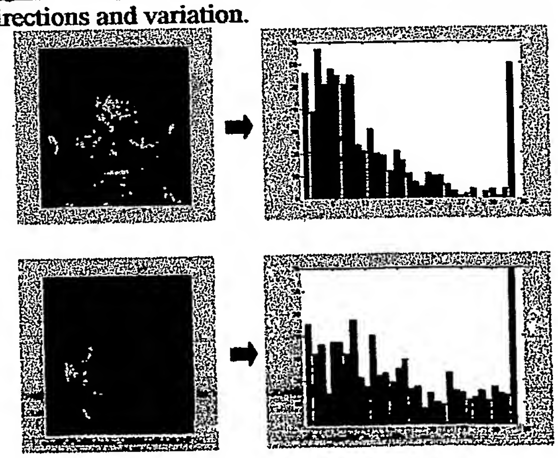


Figure 4. Face VQ Histograms

There are at least two advantages to this method of representing faces with VQ histograms. Because the

VQ Histogram face feature relies on the histogram of the VQ code, it ignores the geometry information, which makes the VQ Histogram feature insensitive to the face position. Also, since the VQ code is generated by the small, minimum intensity subtracted 4-by-4 block, it greatly reduces the effect of lighting on the face. Figure 4~~Figure 4~~ shows a VQ-histogram example.

5. Probabilistic Neural Networks and Modification

The Probabilistic Neural Networks is one of the implementation on Bayes Strategy, which seeking the minimum risk cost based on the Probability Distribution Function (PDF). The standard Probabilistic Neural Networks contains three layers: Input layer, hidden layer and category layer (see Figure 5~~Figure 5~~). The Input Layer will normalize the input vectors and forward the normalized data to Hidden Layer. Hidden layer will then add one node in hidden layer for each training and save the weights as normalized input vectors. (talk about section 4.3)

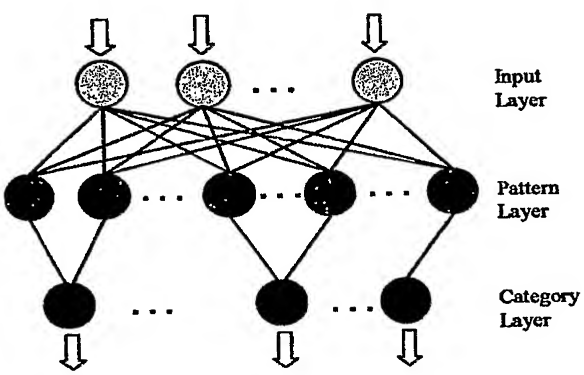


Figure 5. Standard PNN Structure

The category layer will choose the maximum value calculated from hidden layer and calculate the confidence measurement based on the values from hidden layer.

Section 4.1 and 4.2 introduce PNN training and classification and Section 4.3 discuss an adaptive threshold modification on PNN model.

5.1 Probabilistic Neural Network Training

During training, the probabilistic Neural Networks will normalize all the data from input vectors. The normalization calculation is shown below:

$$x'_n = x_n / \sqrt{\sum X_n^2}$$

(2)

where Xn is the input vector and X'n is the normalized input vector.

During each training phase, PNN adds a new node to hidden layer and link the new node with all the nodes in input layer. In order to estimate the PDF of the each category for classification, the weight between the new node and input nodes is assigned with the normalized input vectors; in other words, save all the examples in the link between input nodes and hidden nodes. This will make the PNN generate a Probability Distribution Function during the Classification phase. Then, link the new node in hidden layer to the supervised category. Different to Radial Functions Networks, in PNN, the links between the hidden layer and category layer is not fully connected and doesn't contains any weights.

The training algorithm is listed as below[5]:

1. for X_i, $i = 1, 2, \ldots n$
2. normalize X_i: $x'_{ik} = x_{ik} \bullet \sqrt{\sum X_i^2}$ $k = 1,2,\ldots,d$
3. Assign weights: $W_i = X'_i$
4. if $x_i \in \omega_j$ then $C_{ji} = 1$
5. end

where X_i, $i = 1, 2, \ldots n$ are the input vectors and d is the number of input dimensions, W_i is the weight vector between input nodes and the new hidden node i, and C_{ji} is the link between the new hidden node i and category node j.

Figure 6~~Figure 6~~ shows an example of PNN training with 2 dimension-input vectors. In the figure, we show increasing of the number of hidden nodes. Let the training set X consist of four input vectors: $X = \{x_1\ x_2\ x_3\ x_4\}$, and input vectors $\{x_1\ \ x_2\}$ are supervised to output *category 1*, and input vectors $\{x_3\ \ x_4\}$ are supervised to *category 2*.

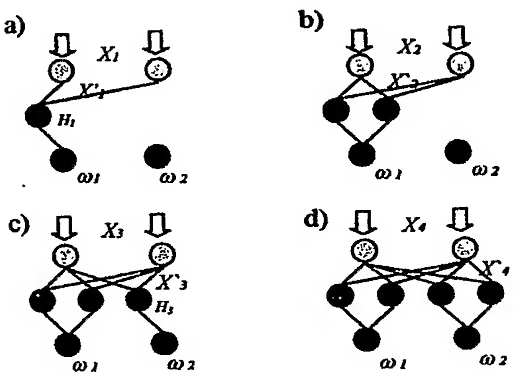


Figure 6. PNN Training Example

For the first training phase shown in Figure 6~~Figure 6~~a, the training data X_1 is normalized into X'_1 and a new hidden node H_1 is added in the hidden layer. The weight of the new hidden node H_1 is assigned using the normalized training data X'_1. Then, the hidden node H_1 is linked to the supervised category node ω_1, which stand for output *category 1* in PNN. For the second training phase for training data X_2, PNN performs the same operations on X_2 as the operations on first training data X_1. As a result of the first and second training, the PNN contains two nodes in hidden layer and they are linked to *category 1* as shown in Figure 6~~Figure 6~~b.

Since the third training data X_3 is supervised to *category 2*, PNN adds a new node H_3 in hidden layer and assign the weights as normalized training data X'_3, but PNN links the new node H_3 to category node ω_2, which is referring the *category 2* as shown in Figure 6~~Figure 6~~c. The PNN performs the same operations for training data X_4. Figure 6~~Figure 6~~d shows a trained PNN with training data X.

5.2 Probabilistic Neural Networks Classification

Because PNN saves all the information from the examples in the hidden layer during training, with some density estimators, PNN can generate a PDF for all the trained categories based on this saved information. There are multiple methods to estimate the PDF, such as Parzen windows [6] and Gaussian Model.

During the Classification process, the probabilistic Neural Networks normalize the input vector as shown in Probabilistic Neural Networks training procedure (see section 4.1). Then in the hidden layer, PNN performs a dot product between the normalized input data and the saved weights as showedn below:

$$Z_i = X' \cdot W_i$$

(3)

where the X' is the normalized input vector and W_i is the weight, Z_i is the output value for each node in hidden layer.

Then PNN performs the following calculation:

$$Z'_i = \sum_{i}^{i \in category_k} \exp[(Z_i - 1)/\sigma^2]$$

(4)

where the σ is the smooth factor and Z_i is the output value for each node in the hidden layer.

In category layer, The PNN makes a classify decision based on the Bayes decision rule, which is shown below in our case:

$$d(\theta = \theta_i) \quad \text{if} \quad Z'_i / n_i > Z'_j / n_j \quad \forall j \neq i$$

(5)

Then, PNN calculates the confidence measurement based on the PDF output from the hidden layer. The nodes in category layer performs the following calculation:

$$C_i = \frac{(Z'_i / n_i)}{\sum Z'}$$

(6)

where the C_i is the confidence measurement for each category, n_i is number of hidden nodes for category i.

5.3 Adaptive Threshold in Probabilistic Neural Networks

In the previous section, we introduced PNN classification procedure. As we mentioned, the PNN can generate the confidence measurements for all categories; however, from formula (5), these confidence measurements are generated by comparing the relative results from the PDF output of the saved examples. Since PNN compares the relative results, sometimes PNN might generate a high confidence output for one category even though the output from PDF for this category is very low.

An example can make the above idea more clear: Figure 7~~Figure 7~~ shows a trained PNN with one-dimension input vector. The PDF is generated based on the saved examples in the hidden layer. In this

case, the input data X_1 is classified as category w_2, and the confidence is calculated by Formula *(5)*. Based on the probability from the PDF on both category w_1 and w_2, $p(x_1|w_1)=0.1$ and $p(x_1|w_2)=0.02$ which generates confidence for input data X_1 of 83%. Also, based on the Formula *(5)*, the input data X_2 is classified as category w_2 with 66% of confidence and input data X_3 is classified as category w_1 with 80% of confidence.

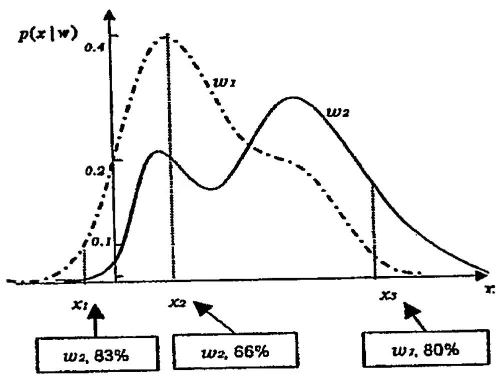


Figure 7. PDF Analysis in PNN: Counter-intuitive confidence problem

The problem is that the input X_1 obtains 83% of confidence, which is larger than 66% of confidence for input X_2, however, the input X_2 is closer to the mean of the PDF of category w_2, which means input X_2 should obtain a higher confidence than input X_1. It is even trickier to explain that the input X_3 has 80% of confidence and in reality cannot provide an 80% confidence.

We introduce a threshold in category layer of PNN to solve the above counter-intuitive confidence problem, which avoids the small PDF output. The category threshold implies a low threshold result in more patterns being declared unknown and an increased number of fine categories.

We update the formula (5) so that it could identify the pattern with low PDF outputs. Formula (7) shows an updated Bayes decision rule with the ability to identify unknown categories:

$$d(\theta=\theta_i) \quad \text{if } Z'_i/n_i > Z'_j/n_j \geq t_i$$
$$d(\theta=unknown) \quad \text{if } Z'_i/n_i > Z'_j/n_j < t_i \qquad (7)$$

$\forall j \neq i$, and t_i is the threshold for category i

Because the PDFs for all categories are different in mean, deviation and maximum conditional probability value, a static threshold might arbitrary erase the right classifier result, which contains low PDF output. Therefore, in order to avoid this problem, different to the technique in [3], we developed an adaptive threshold. The adaptive threshold is chosen based on the percentage of the value of PDFs, which means it calculates the maximum of PDF from examples and for each category a threshold is set based on the their maximum PDF value. As a result, each category will have its own threshold. After any category is selected as classification output, which means its PDF output is the maximum, its PDF output will be compared with its threshold. Only when the category with PDF output is larger than its own threshold, this category can be selected as classification output. Figure 8~~Figure 8~~ shows an updated example with adaptive threshold.

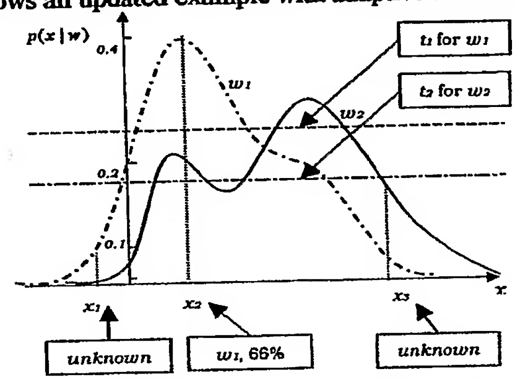


Figure 8. Adaptive Threshold in PDFs

The threshold in figure 8 is set to 70% of the maximum PDF value, and the t1 is the threshold for category 1; t2 is the threshold for category 2. For input vector x1, the $p(x_1|w_1)$ is equal to *0.1*, which is maximum value among the other category PDF values, however, $p(x_1|w_1)$ is still lower than the 70% of maximum PDF value. Thus, input vector x1 is classified as unknown category. For input vector x2, the maximum PDF output among all categories is $p(x_2|w_1)$,which is equal to *0.4*. The $p(x_2|w_1)$ is larger than 70% of the maximum PDF value. PNN then classify the input vector X2 as category 1. PNN perform the same procedure for input vector X3, and because the PDF output for vector X3 is lower than the maximum value of category 2, PNN classify vector X3 as unknown category again.

The result to set thresholds in category layer is to avoid the low confidence classification outputs. Also, as a result of avoiding the low confidence classification, this implementation can detect the unknown categories, which can lower the false alarm of face recognition and detect the unknown faces in our case.

6. Detecting and Online Learning New Faces

Section 4 introduced an Adaptive Threshold Probabilistic Neural Networks (ATPNN) with the ability to recognize the pattern and identify the unknown pattern. Using the face features we introduced in section 3 as ATPNN input, and training the ATPNN with enough face training data, the ATPNN can be set up as a face classifier. The ATPNN based face classifier can recognize the known face and identify the unknown face as well.

We introduce rules to detect the unknown faces for videos based on ATPNN face classifier in section 5.1. Section 5.2 introduces a method to learn online the new faces in ATPNN.

6.1. New Face Detection in Videos

Although the ATPNN can identify the unknown faces, the performance of the ATPNN face classifier for unknown face images is not satisfactory. Based on our experience, the recall rate of the unknown face identification is low with higher threshold, and the false alarm rate of unknown face identification is high with lower threshold. This means that we either miss the unknown faces with a higher threshold or falsely detect the known faces as unknown faces with a lower threshold. Therefore, a post-analysis on ATPNN based face classifier is necessary for detecting the unknown faces.

The solution for detecting the new faces takes advantages of the use of ATPNN and the temporal nature of videos. The ATPNN can provide the identification of unknown faces exactly due to lower recall rate or higher false alarm. As opposed to face images, the video containing faces can provide not only face images but also face sequences in time series. Therefore, we design several conditions to detect new faces, which utilize the advantages of ATPNN and videos.

The conditions to detect a new face are shown below:

1. ATPNN face classifier identifies the face as unknown face
2. Mean of the PDF output is low
3. Variance of the input vectors is small
4. All the above three conditions last for 10 seconds

The condition 1 identifies the input face as an unknown face, and condition 2 evaluates the mean of the PDF output in the face sequence. Condition 3 calculates the distance by performing the standard deviation on the input vectors sequence in order to make sure the input vectors are for the same face. If all three conditions are met within a 10 seconds video clip, we concluded that a new face has appeared in the video.

The algorithm for the above conditions is shown below:

```
if d(θk==unknown) then
  save Xk and Zk into buffer
  if sizeof(buffer) > 10*24 &
             / mean of Zk < th &
             / mean of Xk < thx then
    New Face Found
    Do online Learn
  endif
```

```
Step 1. For face frame k, from ATPNN,
  if d(θk = unknown)
    - Save input vector Xk and PDF output Zk
      to buffer.
    - Go to Step 2.
  else
    - Clear buffer, go to next frame k+1
    - Go to Step 1.

Step 2. In the buffer,
  if buffer size > 10 second * 24
    - Calculate the mean of the PDF output
Zk
    - Calculate the variance of the input
vectors Xk
    - Go to Step 3
  else
    - Go to next frame k+1
    - Go to Step 1

Step 3.
  if the mean and variance is low
    - A new face found, do online learn for
new face
    - Clear buffer
    - Go to Step 1
  else
    - Clear buffer
    - Go to Step 1
```

With the above algorithm, the accuracy of the new face detection becomes very good. Also, because it keeps track the face sequence within 10 seconds, we can avoid the random faces, which happened to be shown in the video.

6.2. Faces Online Learning

Once the algorithm detects a new face, the online learning of the new face is performed. The advantage of PNN is that we do not need to update all the other weights during training [4]. This allows online learning without too many calculations during the updating of weights.

As we described in section 5.1, we store face input vectors in the buffer and we evaluate the variance and mean of these input vectors. In the buffer, the lower

variance input vectors contain more precise information of the new face.

We choose 10 input vectors X_i in the buffer, which contain the low variance from all the input. Then, the PNN learning algorithm is performed for the new input vectors. The procedure of the online training is almost the same as off-line training: normalize the input vector X_i with formula (2). Add a new node into hidden layer i and assign the weights with normalized input vector X'_i. Then, add a new category ω_{new} in category layer and link the hidden node i with the new category ω_{new}.

The algorithm for online learning is shown below:

1. for X_i, $i = 1, 2, \ldots, 10$
2. normalize X_i: $x'_{ik} = x_{ik} \bullet \sqrt{\sum X_i^2}$ $\quad k = 1,2,\ldots,d$
3. assign weights: $W_i = X'_i$
4. $C_{ji} = 1$
5. end

where X_i is the input vectors and d is the number of input dimensions, W_i is the weight vector between input nodes and the new hidden node i, and C_{ji} is the link between hidden node i and the new category node j.

Figure 9~~Figure 9~~a shows a trained ATPNN with 2 faces in database. In this diagram, each hidden node is represented by a face because the nodes save the information of the face during training. Figure 9~~Figure 9~~b shows a PNN after the online learning for a detected new face. In this diagram, the nodes in hidden layer increased and the information for new faces is added into the hidden layer.

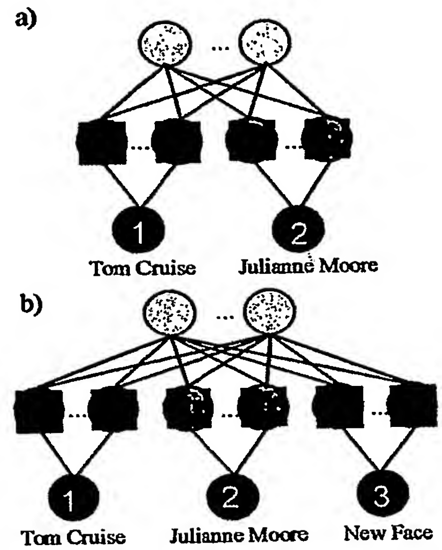


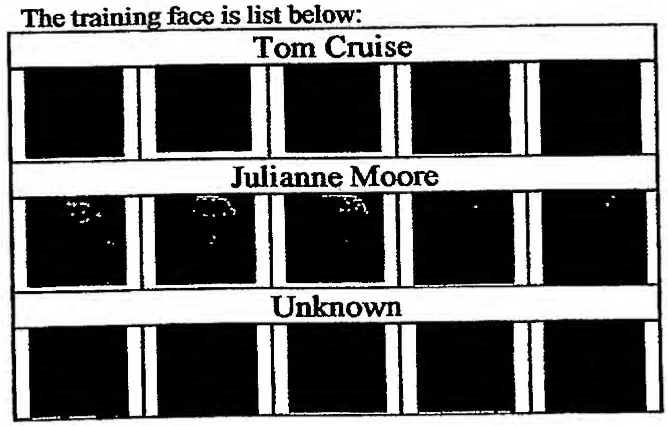
Figure 9. PNN Online Learning/Extension

The information of new face will be stored in hidden layer and category layer of Probabilistic Neural Networks. Therefore, when the "new face" appears again, the probabilistic Neural Networks will recognize this face as known face.

7. Examples

In our implementation, we test the algorithm with a dynamic threshold. If the value of Z' in formula 4 below the threshold, we suppose it is an unknown face and the PNN will output an unknown face ID result. [Generate the curve based on different Th in Matlab].

The training face is list below:



We tested the algorithm on the Movies "Magnolia" (1999) and "Minority Report" (2002). The reason for choosing these two movies is that Tom Cruise appears

in both of them as the main actor. We can train our PNN based on the Tom Cruise's faces in "Magnolia" and use this PNN model to recognize Tom Cruise's face in "Minority Report".

Have some diagram to show the detected "new faces" in here

8. Implementation and Results

We have a test with different threshold in Movie "Magnolia", the first column is generated by Maximum threshold, and the last column is generated by threshold equal to 0.09.

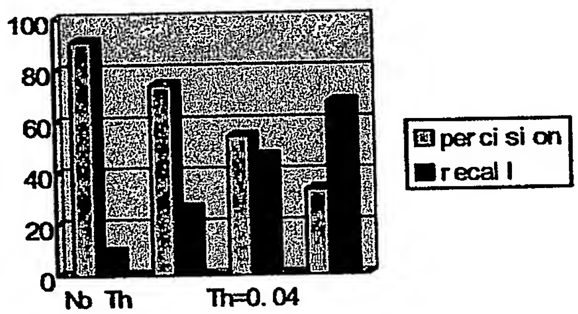


Figure 10 Test results with Static Threshold

Test with adaptive threshold,
And test result for online learning.

[TO BE DONE: Here we will include ROC curves and other results]

9. Conclusion

The main idea is to recognize known faces, detect unknown faces and apply automatic online learning for unknown faces in video. After the online learning, our Classifier could recognize the new (unknown) faces presented before. After the recognition, the Classifier will assign recognized face IDs to the faces.

10. References

[1] Koji Kotani, Chen Qiu, and Tadahiro Ohmi, "Face Recognition Using Vector Quantization Histogram Method", *IEEE International Conference Image Processing, Sept. 2002 PP: 105-108*

[2] Patra P.K, Nayak M., Nayak S.K., Gobbak N.K., "Probabilistic neural network for pattern classification", *IEEE Proceedings of the 2002 International Joint Conference on Neural Networks, May 2002, PP: 1200 –1205.*

[3] Washburne, T.P.; Specht, D.F.; Drake, R.M., "Identification of unknown categories with probabilistic neural networks", *IEEE Proceedings of the 1993 International Conference on Neural Networks, April 1993, PP: 434 -437.*

[4] Specht, D.F., "Probabilistic neural networks for classification, mapping, or associative memory", *IEEE Proceedings of the 1988 International Conference on Neural Networks, July 1988, PP: 525-532.*

[5] R. O. Duda, P. E. Hart, D. G. Stork, "Pattern Recognition, Second Edition". *Wiley & Sons, New York, 2001*

[6] Parzen, E. "On Estimation of Probability density functions and mode" *Ann. Math. Stat., Vol. 33, pp. 1065-1076, Sept. 1962*

[7] M. Turk and A. Pentland, "Eigenfaces for Recognition", *J. Cognitive Neuroscience, Vol. 3, pp. 71-86, 1991*

[8] P. Viola and M. Jones, "Rapid Object detection using a boosted cascade of simple features", IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, 2001

[9] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection", IEEE ICIP, Vol. 1, pp. 900-903, Sep. 2002

[10] M. Jones, Personal Communication, 2003

[11] S.Z. Li et al, "Floatboost learning for classification", Proceedings of the 16th Annual Conference on NIPS, Vancouver, Canada, Dec. 2002

[12] R. Schapire and Y. Freund, "Improved boosting algorithms using confidence-rated predictions", Machine Learning, 37(3), pp. 297-336, 1999

ONLINE FACE RECOGNITION SYSTEM FOR VIDEOS BASED ON MODIFIED PROBABILISTIC NEURAL NETWORKS WITH ADAPTIVE THRESHOLD

Jun Fan, Nevenka Dimitrova and Vasanth Philomin

Philips Research USA
Briarcliff Manor, NY 10510
junfan@ee.columbia.edu,
nevenka.dimitrova@philips.com

Philips Research Aachen,
Germany
vasanth.philomin@philips.com

ABSTRACT

Video retrieval in consumer applications demands high level semantic descriptors such as people's identity. The problem is that in a variety of videos such as *home videos, Hollywood content, TV broadcast content, mobile phone videos faces are not easy to recognize. Even more, a closed system trained to recognize only a predetermined number of faces will become obsolete very easily.* We developed an online-learning face recognition system for a variety of videos based on Adaptive Threshold Probabilistic Neural Networks (ATPNN). . This face recognition system can detect and recognize known faces, as well as automatically detect unknown faces and train the unknown faces online into new face classifiers such that this "unknown face" can be recognized if it appears again. MPNN is a variant of PNN with adaptive thresholding. on the category (output) layer of aProbabilistic Neural Network (PNN) in order to detect unknown categories of input data. The PNN feed-forward training makes the online training very fast because adding new faces does not require retraining of the known categories. Our results show that off-line and on-line learning yield equivalent results. The real added benefits are: 1) we can build open systems, and 2) PNN makes because there is no retraining required for known faces.

1. INTRODUCTION

Most face recognition systems are trained on a fixed number of faces that are known in advance. These systems will only recognize the faces with known models and the face database cannot be updated during the classification procedure. In this respect these systems are very limited once they are placed in operation. For example, they will work for surveillance systems, which have to recognize all employees of a company and alert to any intruders, or an airport surveillance system that is trained to recognize known terrorists. However, in the area of home video, TV broadcast video, wearable video, in addition to the known people there is a need to recognize the new people appearing with each new video. In home videos for example, if a system is trained to recognize only family members then a visitor is labeled as "other" or "unknown". Of course there are travel videos with many new faces that are transient. A system that categorizes images and videos based on people presence has to distinguish all these categories of important and unimportant faces. Moreover, the system has to be flexible enough to incorporate and retain important faces.

Our approach can automatically detect the new faces and extend the database based on the new faces. Our online learning system can learn features of new, reoccurring faces and store corresponding models of new faces for future use. Also, our approach can generate a confidence measurement for each recognized face in the database and sort the candidate by the confidence measurement, which make post-processing easier.

2. SYSTEM ARCHITECTURE

Figure 1 shows our face recognition system architecture. There are two approaches to bootstrap the system: 1) Initial database has a limited number of faces, and 2) Initial database is empty. If the system is first trained on the initial database, we can gain higher recognition accuracy on the initial database. This method is similar to our human perception of known faces and incorporation of new faces. The system has a training phase and classification phase just like any other face recognition system (depicted with a dashed line). However, the important aspect here is that there is a feedback arrow to the training face for unknown faces. The persistent (reoccurring) faces become new sample faces for the online training (dotted ellipse in figure 1).

During the training phase, the system reads face examples for each face (actor/character) and trains the Probabilistic Neural Networks (PNN) [4][2] based on the features of these faces We choose Vector Quantization Histogram features as face features [1]. During the classification phase, the system will decode the MPEG video file into video frames first. For each frame, we use a variant of the face detector described in [8]. If there is a face found by the face detector, the face segment is forwarded to the PNN based Face Classifier. A confidence measurement for each face ID is generated by PNN. Based on an adaptive thresholding of the confidence values and a set of conditions the system determines if the face is

known or unknown. Persistent unknown faces are evaluated and forwarded to online learning phase. After we have the confidence measurement for each Face ID, we can easily choose the Face ID with the maximum confidence measurement as the output from Face Classifier by using a Winner takes all principle.

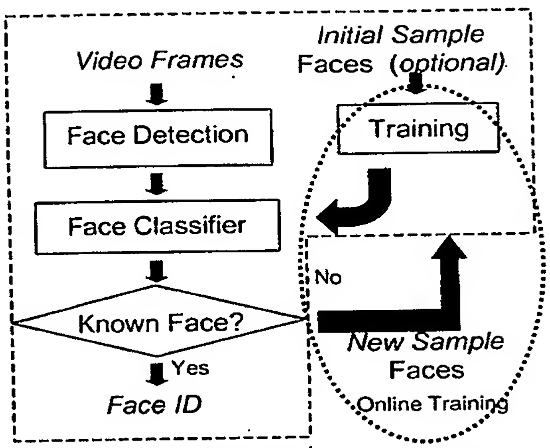


Figure 1. Face Recognition System Architecture

3. FACE DETECTION

This section briefly describes the face detection algorithm used in our framework. In [8], Viola and Jones applied the popular AdaBoost [12] learning technique to the problem of rapid object detection. They used an attentional cascade of strong classifiers that consisted of a set of computationally efficient binary features (also called weak classifiers). Each round t of boosting added a single feature h_t to the current set of features by minimizing:

$$Z_t = \sum_i D_t(i)\exp(-\alpha_t y_i h_t(x_i)),$$

where $D_t(i)$ is the weight on example x_i at round t, $y_i \in [-1, 1]$ is the target label of the example, α_t is the influence of this weak hypothesis on the strong classifier and $h_t()$ is the weak binary hypothesis restricted to [-1, 1]. In our variant, we use boosting stumps (decision trees that partition the domain into two pieces and yield a prediction for each partition) as the weak classifiers, which results in α_t being folded into h_t, thereby allowing the weak hypotheses to have a range over all $\Re$ rather than the restricted range [-1, +1]. The prediction values for the left and right partitions that minimize Z_t above are:

$$c_{left} = \frac{1}{2}\ln(\frac{W_+^{left}+\varepsilon}{W_-^{left}+\varepsilon}); c_{right} = \frac{1}{2}\ln(\frac{W_+^{right}+\varepsilon}{W_-^{right}+\varepsilon}),$$

where the W's denote the weight of the examples that are assigned to the left or right partition with true labels "positive" or "negative". The predictions are also smoothed with the term ε to avoid numerical problems caused by large predictions. From these prediction values, we can greedily choose the splitting criterion for the decision tree (dropping the subscript t) as

$$Z = 2(\sqrt{W_+^{left}W_-^{left}} + \sqrt{W_+^{right}W_-^{right}})$$

rather than the Gini index or an entropic function [12]. A few variants [9][11] of the learning algorithm described in [8] have been proposed recently. These algorithms reduce the training error (i.e. error in the training set) during training and count on the generalization performance of AdaBoost that is rigorously proved in [12]. It is our experience that using a validation set during training as in [8][10] yields the most effective cascades with fewer features. This is due to the fact that we get multiple hits around each face while scanning the validation set and we can pick the strong classifier threshold as high as possible in order to retain just one hit, thereby eliminating more false alarms in the process. However, one must ensure that this threshold is not chosen too high so as to miss too many positive training images. In addition, we just scan the validation set once (rather than several times as in [10]) to adjust the strong classifier threshold as each weak classifier is added to the current cascade. We do this by keeping track of the rectangles and their corresponding last stage sums that pass through all but the penultimate stage of the current cascade (for the first stage, this amounts to keeping track of all rectangles scanned and their corresponding sums). We use around 4000 positive samples and 5000 negative samples for training each stage of the cascade where the negative samples for each stage are the false positives obtained by scanning the current cascade on an image set with no faces. Our validation set consists of around 200 faces.

4. ONLINE FACE RECOGNITION

This section describes the online face recognition algorithm used in our framework. Firstly, we introduce a face classifier based on Adaptive Threshold Probabilistic Neural Networks (ATPNN), which develop from Probabilistic Neural Networks [4]. We have two reasons for choosing PNN as our face classifier:. 1) we can measure the outputs confidence based on the preset threshold, and 2)PNN is a feed forward training model, which means it is not necessary to train the existing links in PNN when adding new node in the category (output) layer. Then, we introduce the conditions we used for new face detecting and the online learning algorithm for new faces in the later sub-sections.

4.1. Adaptive Threshold Probabilistic Neural Networks

The ATPNN is developed from Probabilistic Neural Networks, which Specht, D.F first introduced in [4]. The PNN is one of the implementation on Bayes Strategy, which seeking the minimum risk cost based on the Probability Distribution Function (PDF). The Bayes Decision rule used in PNN is shown below:

$$d(\theta = \theta_i) \text{ if } Z_i > Z_j \quad \forall j \neq i \qquad (1)$$

where θ is the output category, Z is the PDF output of each input vector.

The PNN can generate a confidence measurement by comparing the relative results from the PDF output of the saved examples. However, this makes PNN generate a high confidence output for one category even though the output from PDF for this category is very low. For example, in Figure 2, a trained PNN with one-dimension input vector is shown. The PDF is generated based on the saved examples in the hidden layer. Without the thresholds t, the input vector x1 can be classified as category w1, with around 80% high confidence. However, the PDF output for the input vecter x1 is low enough to be identified as unknown category.

By adding a threshold in the category layer of PNN, the PNN can identify unknown categories [3] and also avoid identifying low PDF output vectors. After the modification, the Bayes Decision rule is updated as below:

$$d(\theta = \theta_i) \qquad \text{if } Z_i > Z_j \geq t \quad \forall j \neq i$$

$$d(\theta = unknown) \text{ if } Z_i > Z_j < t \quad \forall j \neq i \qquad (2)$$

where t is the threshold in category layer.

The ATPNN based face classifier can recognize the known face and identify the unknown face as well. If outputs for all faces (nodes in the category layer) are below the threshold, we assume the input face is an unknown face. Figure 2 shows the strategy

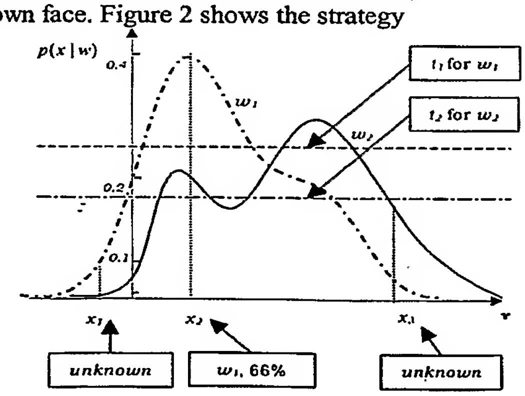


Figure 2. Adaptive Threshold for PDFs in ATPNN

4.2. New Face Detection in Videos

The solution for detecting the new faces takes advantage of the use of ATPNN and the temporal nature of videos. The ATPNN can provide the identification of unknown faces exactly due to lower recall rate or higher false alarm. As opposed to face images, the video containing faces can provide not only face images but also face sequences in time series. Therefore, we design several conditions to detect new faces, which utilize the advantages of ATPNN and videos.

The conditions to detect a new face are shown below:

1. ATPNN face classifier identifies the face as unknown face
2. Mean of the PDF output is low
3. Variance of the input vectors is small
4. All the above three conditions last for n (e.g. n=10) seconds

The condition 1 identifies the input face as an unknown face, and condition 2 evaluates the mean of the PDF output in the face sequence. Condition 3 calculates the distance by performing the standard deviation on the input vectors sequence in order to make sure the input vectors are for the same face. If all three conditions are met within the n seconds video clip, we concluded that a new face has appeared in the video. This is a simple use of "memory". However if a face appears many times in a video for very short periods (high-cut rate in a conversation for instance) then we need to employ accumulative memory w here a face of a stranger is learned over time (e.g. reapeating faces in home video that appear at different social gatherings.

4.3. Faces Online Learning

Once the algorithm detects a new face, the online learning of the new face is performed [what method of on-line learning is used?]. The advantage of PNN is that we do not need to update all the other weights during training [4]. This allows online learning without too many calculations during the updating of weights.ok!

As we described in section 5.1, we store face input vectors in the buffer and we evaluate the variance and mean of these input vectors. In the buffer, the lower variance input vectors contain more precise information of the new face.

We choose 10 input vectors X_i in the buffer, which contain the low variance from all the inputs (i.e. the closest closest to the average in the buffer). Then, the PNN learning algorithm is performed for the new input vectors. The procedure of the online training is almost the same as off-line training: normalize the input vector X_i with formula (2) For every X_i add a new node into the hidden layer and initialze the weights of the node to the normalized input vector X'_i. Then, add a new category ω_{new} in the category layer and link the added hidden nodes to the new category ω_{new}.

The algorithm for online learning is shown below:

1. for X_i, $i = 1, 2, \ldots, 10$
2. normalize X_i: $x'_{ik} = x_{ik} \bullet \sqrt{\sum X_i^2} \quad k = 1,2,\ldots,d$
3. assign weights: $W_i = X'_i$
4. $C_{ji} = 1$
5. end

where X_i is the input vectors and d is the number of input dimensions, W_i is the weight vector between input nodes

and the new hidden node i, and C_{ji} is the link between hidden node i and the new category node j.

Figure 3a shows a trained ATPNN with 2 faces in database. In this diagram, each hidden node is represented by a face because the nodes save the information of the face during training. Figure 3b shows a PNN after the online learning for a detected new face. In this diagram, the nodes in hidden layer increased and the information for new faces is added into the hidden layer.

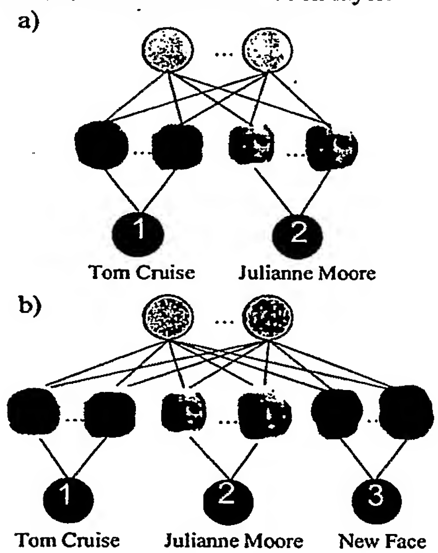


Figure 3. Adding a new face using PNN Online Learning/Extension

The information about the new face will be stored in hidden layer and category layer of the Probabilistic Neural Network. Therefore, when the "new face" appears again, the probabilistic Neural Networks will recognize this face as known face. Here of course it is possible to continue online training to reinforce the new face classifier. However exploring how much weight will a new face example have in re-enforcing the classifier is really part of our future research.

5. IMPLEMENTATION AND RESULTS

We tested 4 genres of videos:, movies, News Video, Video Conference and Home Video. The experimental resultare shown in Table 1.

Video Category	Min	# of Faces	Offline		Detected	Online	
			Hit	FP		Hit	FP
Movies	303	39	82 %	19 %	30	77 %	32 %
News	22	24	93 %	27 %	9	81 %	18 %
Conference	45	6	91 %	5 %	6	90 %	6%
Home Video	28	6	74 %	24 %	2	52 %	45 %

Table 1. Experiments result on different genre

In Table 1, the second column labeled as "Min" refers to the minutes of the videos in total, the third column labeled as "Faces" refers to the number of detectable faces. "Hit" means the hit ratio and "FP" means the false positive. The column labeled as "Online" means the number of faces that has been detected and online trained in PNN. We should note here that by design we chose to detect new faces that are persistent for more than 10 seconds.

From Table 1, we see there are 24 faces in the 22 minutes News video, however, the algorithm is instructed to only learn online 9 faces out of 22 faces. This is because most of the new faces in News video are short in length, and the algorithm ignores the new face before adding it into face database For the movies we during the online learning experiment, we initialize the training set with 4 actors and for each actor, we used 5 face samples.

Movie	Actor	Offline		Online	
		Hit	FP	Hit	FP
Magnolia	Tom Cruise	86%	7%	81%	11%
	Julianne Moore	91%	15%	72%	26%
	Philip B. Hall	67%	19%	65%	24%
	Jeremy Blackman	81%	11%	69%	16%

Table 2. Experiment results for particular actors

Table 2 shows the experiment results for particular actors. Recognition result of off-line learning and on-line learning are shown in this table. We can find the result of Online learning is comparable to the offline learning result.

6. CONCLUSIONS

Open systems for detection and recognition of high level semantic descriptors are going to be increasingly valuable in consumer's world of multimedia content explosion. Once in operation the system should be able to learn and adapt just like babies learn with time to recognize the faces of their parents, close relatives, friends and keep expanding. In this paper we introduced an online face recognition system that uses a variant of the Probabilistic Neural Networks. The main goal is to recognize known faces, detect unknown faces and apply automatic online learning for unknown faces in video. After the online learning, our Classifier could recognize the new (unknown) faces presented before. After the recognition, the Classifier will assign recognized face IDs to the faces. Our initial results are very promising.

In the future we would like to explore this concept further to include intermittently persistent faces (e.g. a presidential candidate is shown more often until he/she becomes important). There are different forms of memory that can enrich the system to attain more human-like recognition capabilities.

7. REFERENCES

[1] Koji Kotani, Chen Qiu, and Tadahiro Ohmi, "Face Recognition Using Vector Quantization Histogram Method", *IEEE ICIP, Sept. 2002 PP: 105-108*

[2] Patra P.K, Nayak M., Nayak S.K., Gobbak N.K., "Probabilistic neural network for pattern classification", *IEEE 2002 IJCNN, May 2002, PP: 1200 –1205.*

[3] Washburne, T.P.; Specht, D.F.; Drake, R.M., "Identification of unknown categories with probabilistic neural networks", *IEEE 1993 IJCNN International Conference on Neural Networks, April 1993, PP: 434 -437.*

[4] Specht, D.F., "Probabilistic neural networks for classification, mapping, or associative memory", *IEEE 1988 ICNN, July 1988, PP: 525-532.*

[5] R. O. Duda, P. E. Hart, D. G. Stork, "Pattern Recognition, Second Edition". *Wiley & Sons, NY, 2001*

[6] Parzen, E. "On Estimation of Probability density functions and mode" *Ann. Math. Stat., Vol. 33, pp. 1065-1076, Sept. 1962*

[7] M. Turk and A. Pentland, "Eigenfaces for Recognition", *J. Cognitive Neuroscience, Vol. 3, pp. 71-86, 1991*

[8] P. Viola and M. Jones, "Rapid Object detection using a boosted cascade of simple features", IEEE CVPR, Kauai, HI, 2001

[9] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection", IEEE ICIP, Vol. 1, pp. 900-903, Sep. 2002

[10] M. Jones, Personal Communication, 2003

[11] S.Z. Li et al, "Floatboost learning for classification", Proceedings of the 16th Annual Conference on NIPS, Vancouver, Canada, Dec. 2002

[12] R. Schapire and Y. Freund, "Improved boosting algorithms using confidence-rated predictions", Machine Learning, 37(3), pp. 297-336, 1999